ELE 336 Microprocessors
Section 21 & 22

Syllabus

Hacettepe University
Department of Electrical and Electronics Engineering
ELE 414 Microprocessor and Programming II All Sections

Tuesday
Instructors: Assoc Prof. Ali Ziya Alkar
Office Hours: Friday 09:00-11:00
e-mail: alkar@hacettepe.edu.tr

Prerequisite: In order to take this course you should have taken the prerequisite course ELE 118 and have done well.

TextBooks:
Brey, The Intel Microprocessors, Prentice Hall, 5thEdition.
Gaonkar, Microprocessor Architecture Programming and Apps /Prentice Hall. Besides the other aspects of the 8085 programming we will talk about the programmable 8085 peripherals and data transfer.

Useful Books:
M. A. Mazidi &. G. Mazidi,"The 80x86 IBM PC and Compatible Computers", Prentice Hall,2000.
Antonakos, An Introduction to the Intel Family of Microprocessors, Prentice Hall, 1999
K.R. Irvine, Assembly Language for Intel Based Computers, Prentice Hall,1999.
W. A. Triebel and A. Singh, The 8088and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications” Prentice Hall, 2000
Flynn, Computer Architecture Pipelined and Parallel Processor Design
Computer Architecture and Logic Design, Thomas Bartee, McGraw Hill
and in combination with other computer architecture books available.

Syllabus

LAB for the course:

Essential Programs for the course:

The DEBUG command on DOS.

MASM Assembler, CODEVIEW and emu8086v103.zip.

See lab page for more information

Midterm %45, Final %50,Homework %5

Attempts of cheating in Homeworks and Lab-Works will NOT be tolerated.

No exceptions.

Attendance: Required in ALL course hours and ALL LAB hours

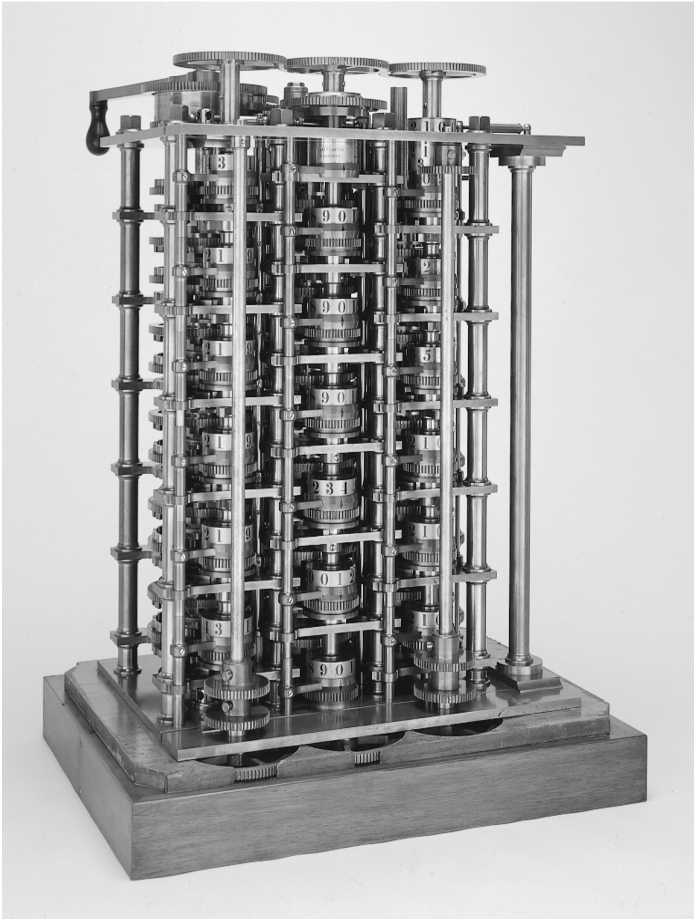
WEEKS
1. Introduction to Microcomputers and Microprocessors,
2. 80x86 Processor Architecture, Internals, Registers, Flags, Segments
3. 8088/8086 Instruction Set, Machine Codes, Addressing Modes, basic instructions, data transfer instructions
4. 8088/8086 Instruction Set, Machine Codes, Addressing Modes,arithmetic logic instructions
5. 8088/8086 Microprocessor, instruction set, program control instructions
6. The 8088 and 8086 Microprocessors programming, BIOS and DOS interrupts.
7. Memory and Memory Interfacing for 8088
8. Memory and Memory Interfacing for 80x86

9. Midterm
10. Input/Output Interface Circuits and Peripheral Devices
11. Input/Output Interface Circuits and Peripheral Devices
12. I/O interfacing with 8255
13. Serial Data Communication and 16450/8250/8251 chips
14.

Week 1

Introduction to Microcomputers and Microprocessors, Computer Codes, Programming, and Operating Systems

First Computer



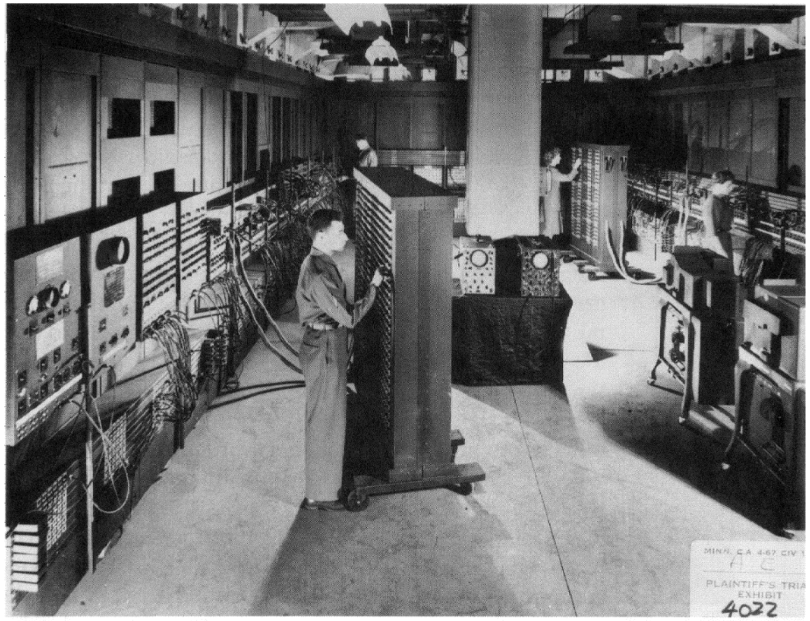
• 1832 Babbage mechanical machine to calculate the navigation tables for the Royal Army, U.K.

The Babbage Difference Engine (1832)

25,000 parts

cost: £17,470

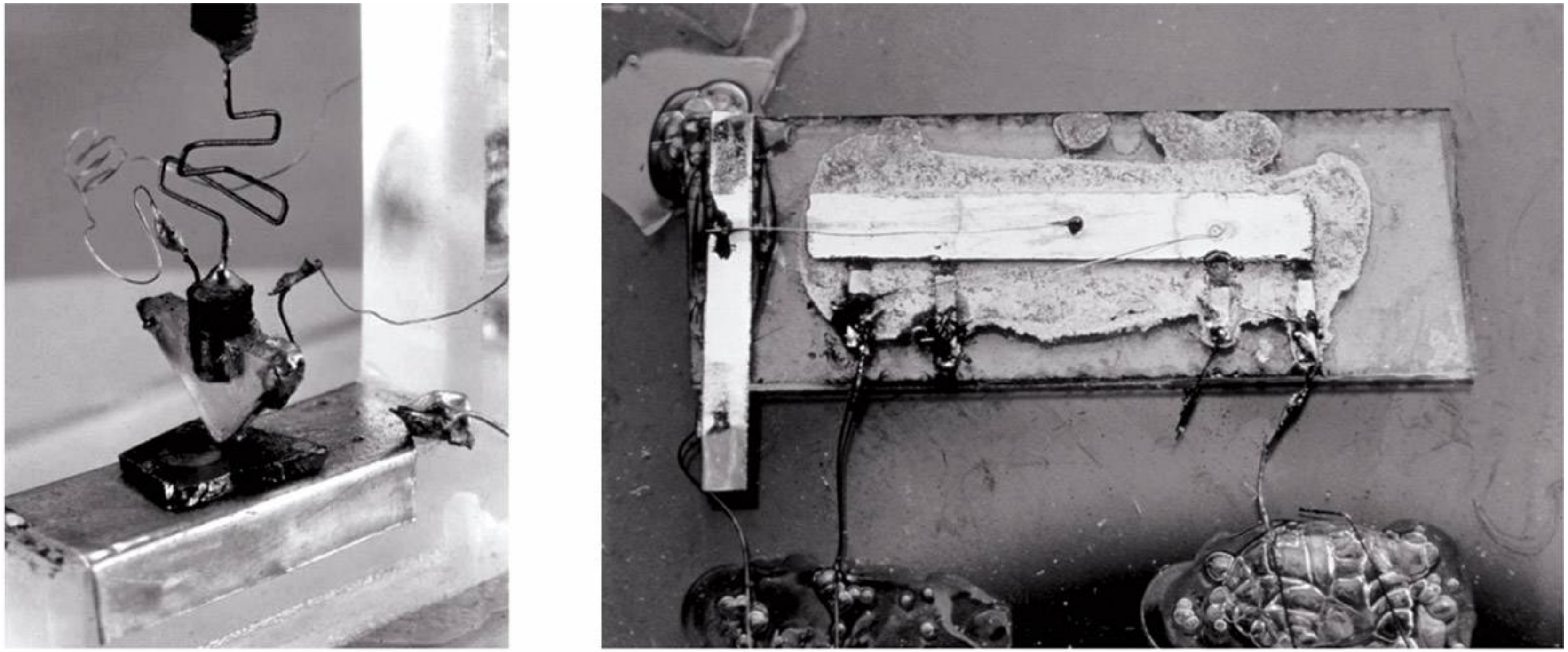
ENIAC



• Vacuum tube based
• “BIG BRAIN”
• ENIAC
–1,800 sq. Feet area
–30 ton
– 18000 vacuum tubes
• Application: IInd WW

- 1943 First electronic computer is used to decode the German Army secret codes, coded by the enigma machine: Colossus,
- 1946 First General Purpose computer: ENIAC 17000 vacuum tubes, 500 miles of wire 30 tons, 100 000 ops per sec.@ U.of Penn

First Transistor



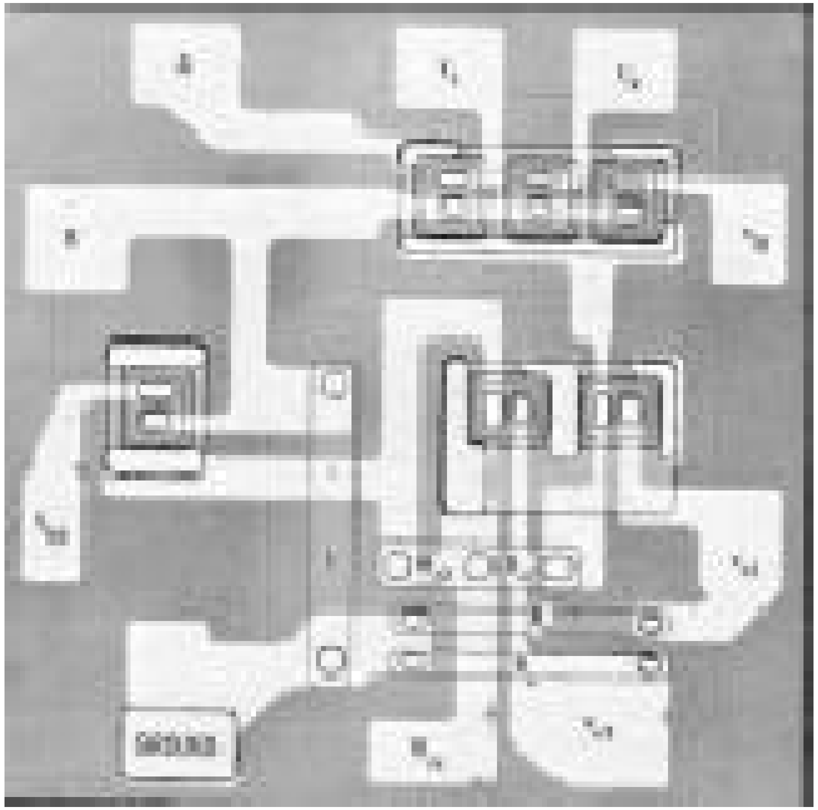
(a) (b)

FIG 1.2 (a) First transistor (Courtesy of Texas Instruments.) and (b) first integrated circuit. (Property of AT&T Archives. Reprinted with permission of AT&T.)

Bell Labs 1946

First Integrated Circuit (IC)

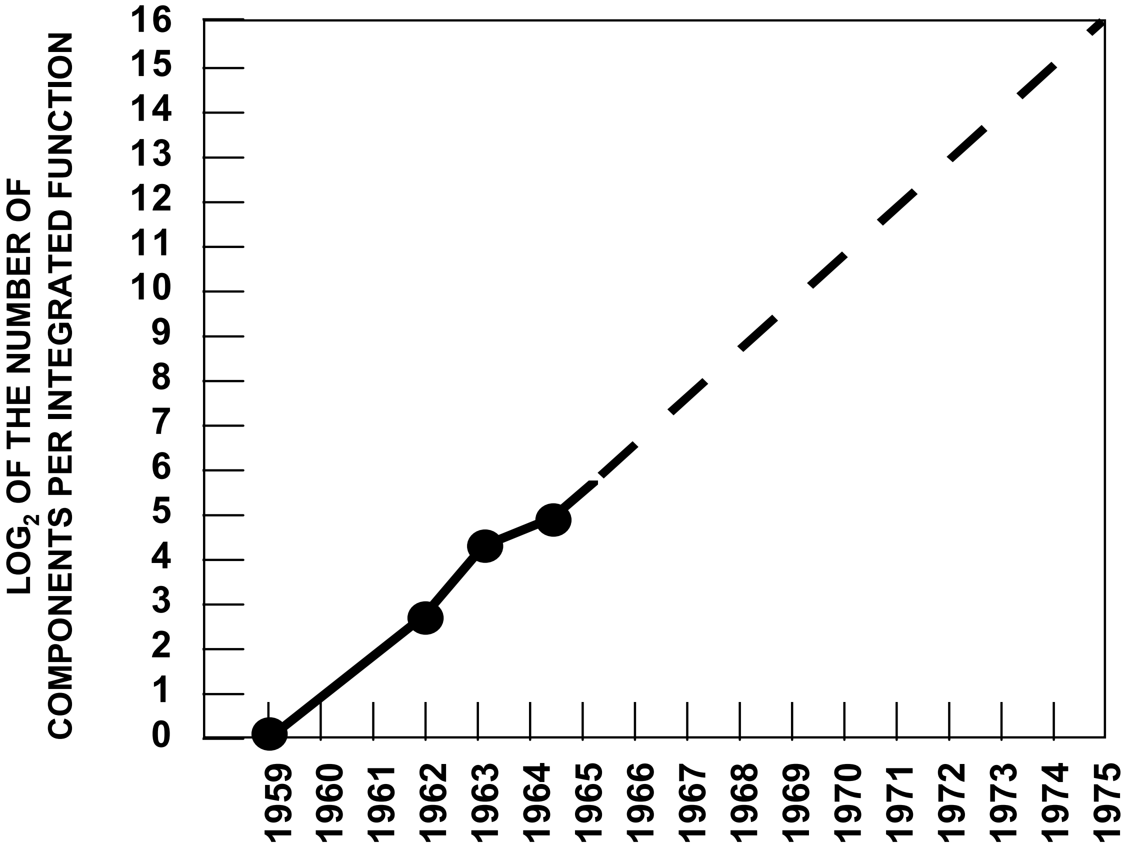
- 1958 Invention of the IC by Jack Kilby at Texas Instruments



Bipolar logic
1960's

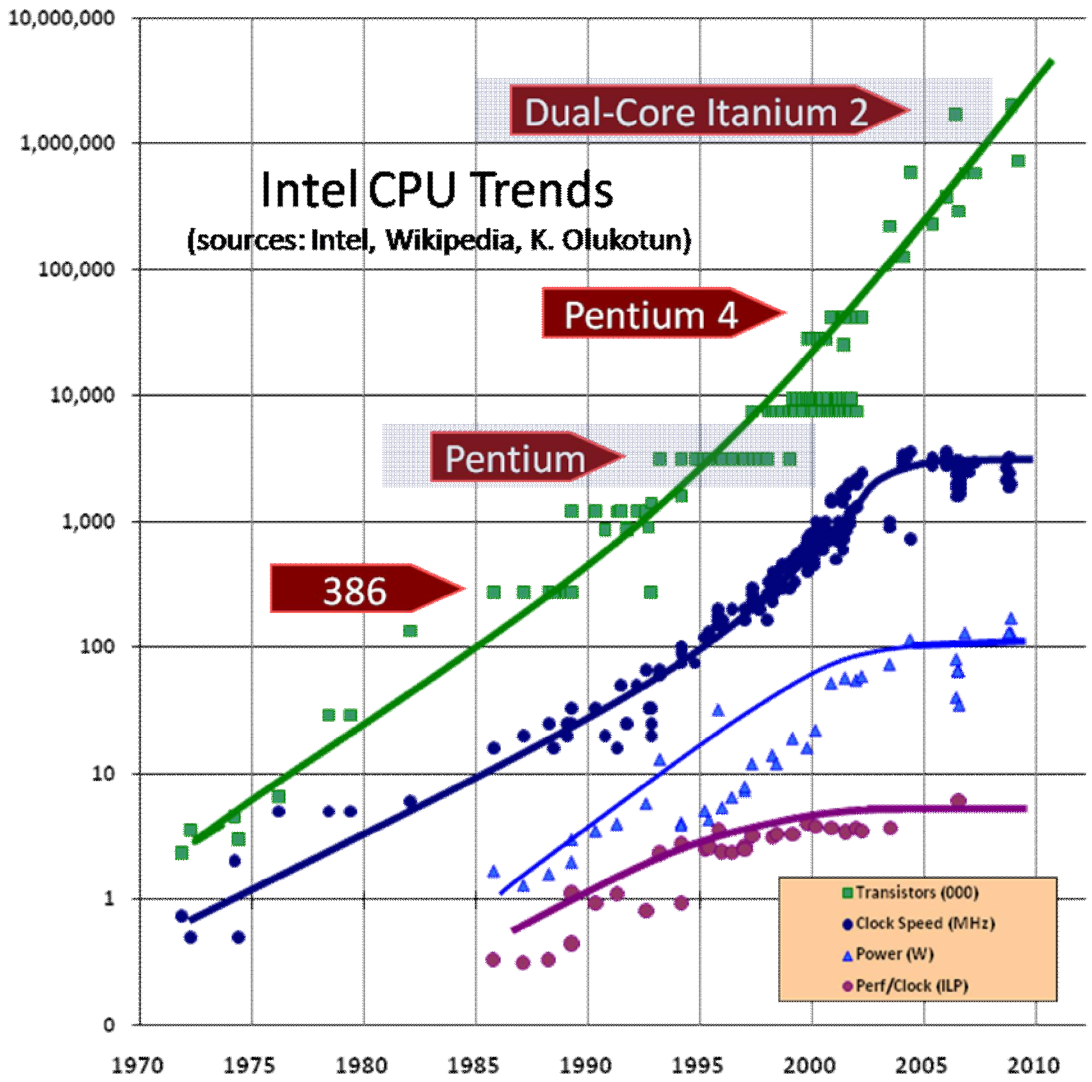
ECL 3-input Gate
Motorola 1966

Moore's Law

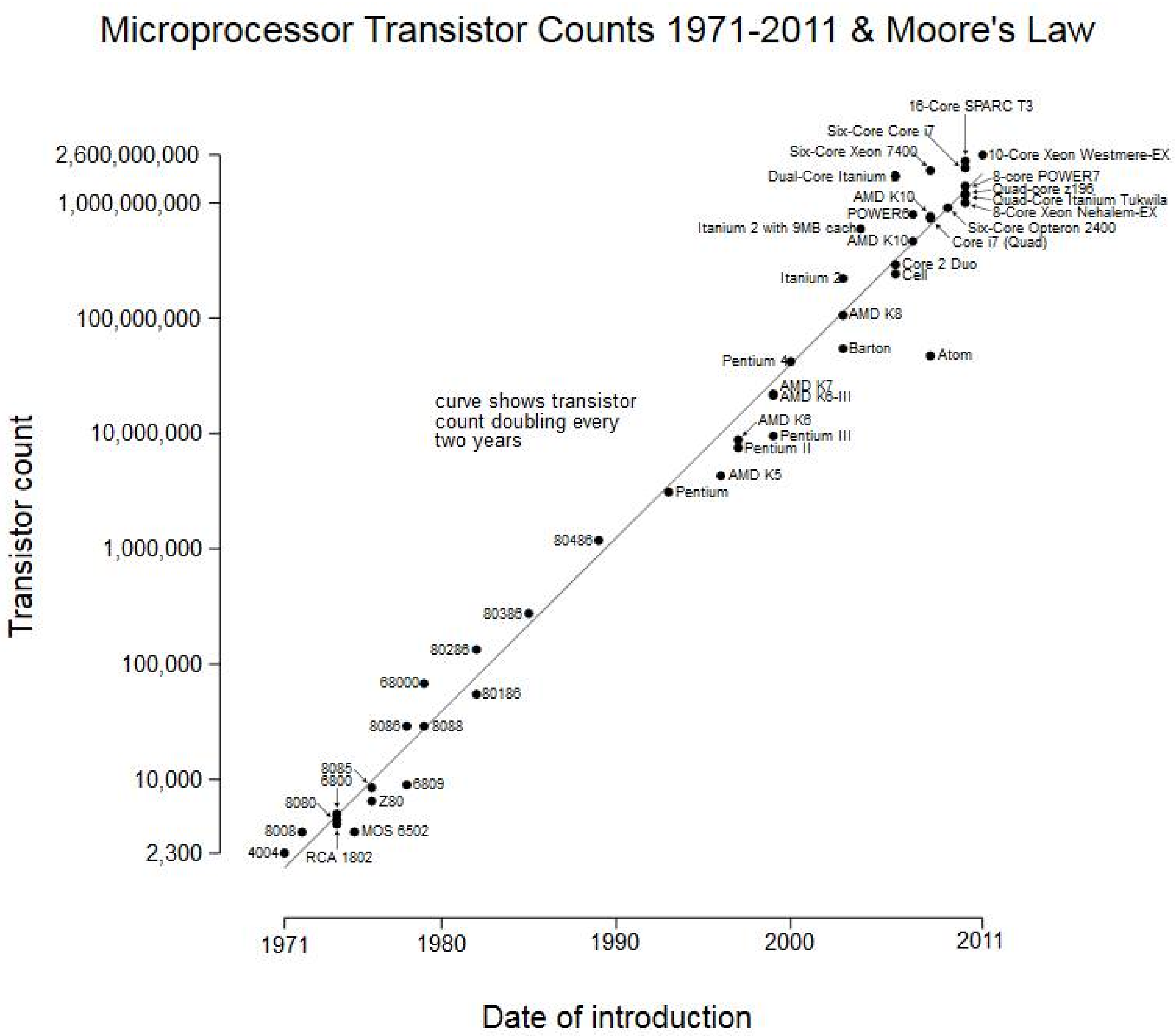


Intel's Founder Gordon Moore 19 April 1965, Electronics

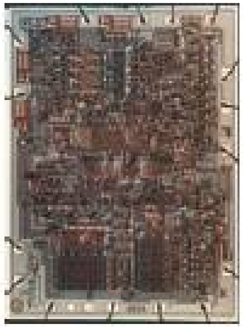
Performance concerns

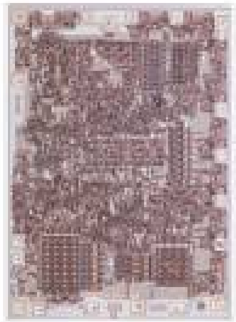


Change over the years



Change in Transistors



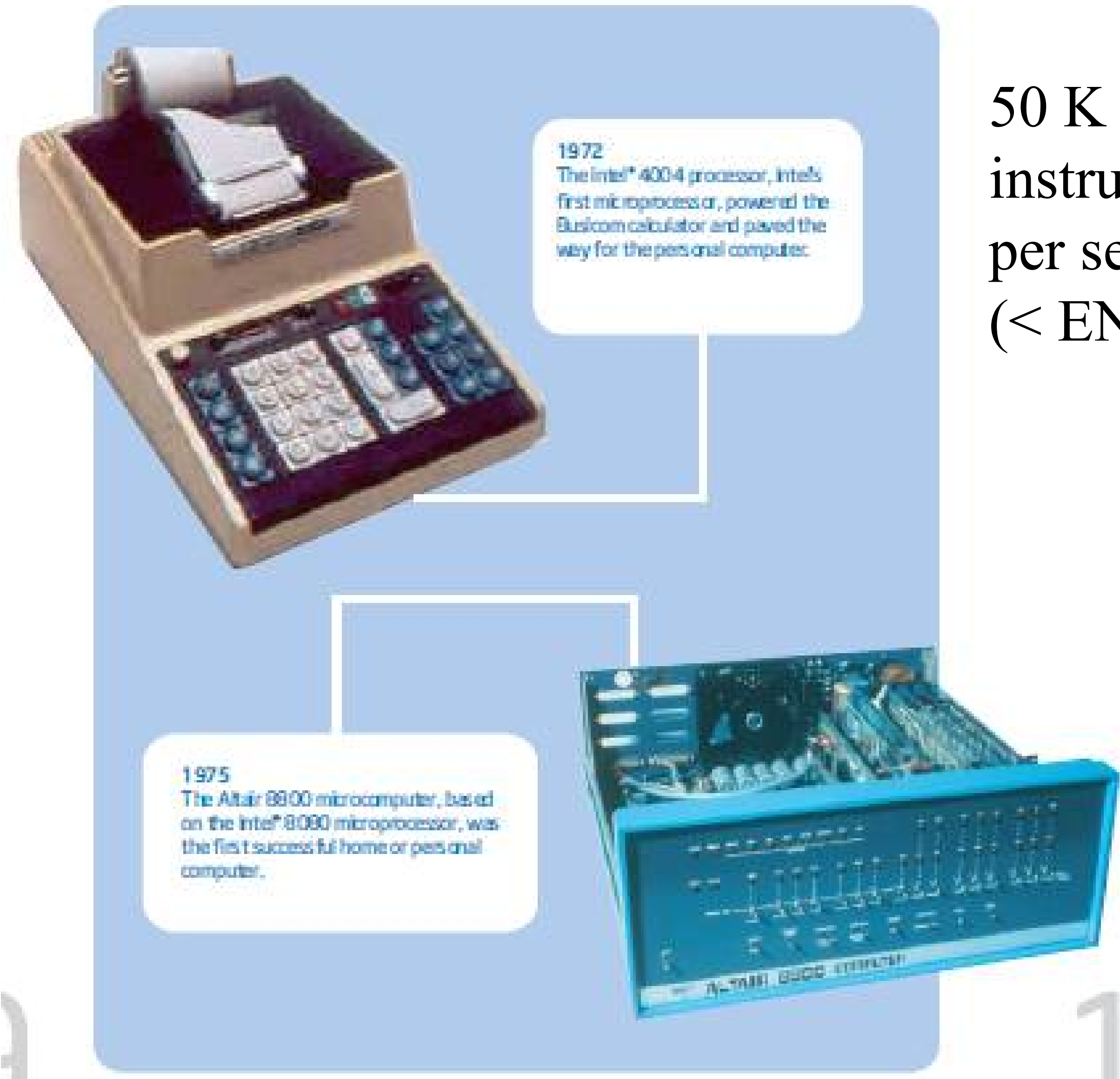


Intel® 4004 processor
Introduced 1971
Initial clock speed
108 KHz
Number of transistors
2,300
Manufacturing technology
10μ

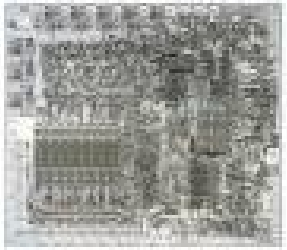
The groundbreaking Intel® 4004 processor was introduced with the same computing power as ENIAC.

Intel® 8008 processor
Introduced 1972
Initial clock speed
500-800 KHz
Number of transistors
3,500
Manufacturing technology
10μ

The Intel® 8008 processor was twice as powerful as the Intel® 4004 processor.

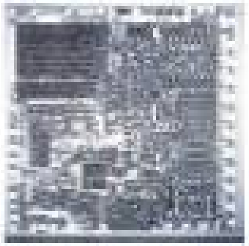


50 K instructions per second (< ENIAC!).



Intel® 8080 processor
Introduced 1974
Initial clock speed
2 MHz
Number of transistors
4,500
Manufacturing technology
6μ

The Intel® 8080 processor made video games and home computers possible.



Intel® 8086 processor
Introduced 1978
Initial clock speed
5 MHz
Number of transistors
29,000
Manufacturing technology
3μ

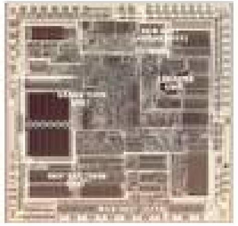
The Intel® 8086 processor was the first 16 bit processor and delivered about ten times the performance of its predecessors.

Intel® 8088 processor
Introduced 1979
Initial clock speed
5 MHz
Number of transistors
29,000
Manufacturing technology
3μ

A pivotal sale to IBM's new personal computer division made the Intel® 8088 processor the brains of IBM's new hit product—the IBM PC.

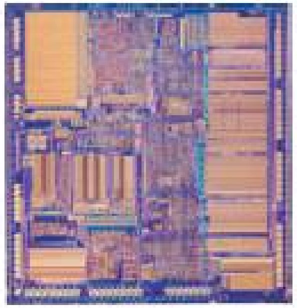
1976
An operator in an early bunnysuit shows how a 4-inch wafer is prepared for a positive acid spin.

1981
The Intel® 8088 microprocessor was selected to power the IBM PC.



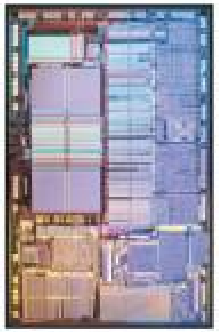
Intel® 286 processor
Introduced 1982
Initial clock speed
6 MHz
Number of transistors
134,000
Manufacturing technology
1.5μ

The Intel® 286 was the first Intel processor that could run all the software written for its predecessor.



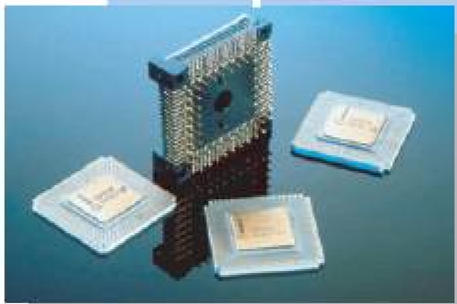
Intel386™ processor
Introduced 1985
Initial clock speed
16 MHz
Number of transistors
275,000
Manufacturing technology
1.5μ

The Intel386™ processor could run multiple software programs at once and featured 275,000 transistors—more than 100 times as many as the original Intel® 4004.



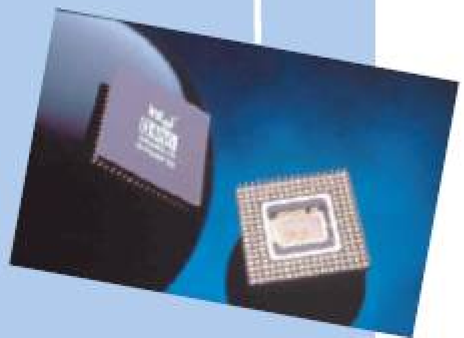
Intel486™ processor
Introduced 1989
Initial clock speed
25 MHz
Number of transistors
1,200,000
Manufacturing technology
1μ

The Intel486™ introduced the integrated floating point unit. This generation of computers really allowed users to go from a command level computer into point and click computing.



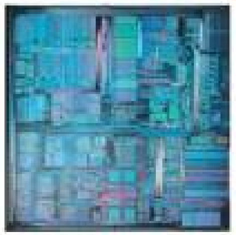
1982
Within 6 years of its release, an estimated 15 million 286-based personal computers were installed around the world.

1989
The National Academy of Engineering named the microprocessor one of ten outstanding engineering achievements for the advancement of human welfare.



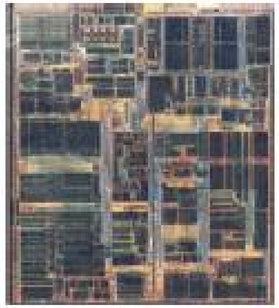
Intel® Pentium® processor
Introduced 1993
Initial clock speed
66 MHz
Number of transistors
3,100,000
Manufacturing technology
0.8µ

The Intel® Pentium® processor, executing 112 million commands per second, allowed computers to more easily incorporate "real world" data such as speech, sound, handwriting and photographic images.



Intel® Pentium® Pro processor
Introduced 1995
Initial clock speed
200 MHz
Number of transistors
5,500,000
Manufacturing technology
0.6µ

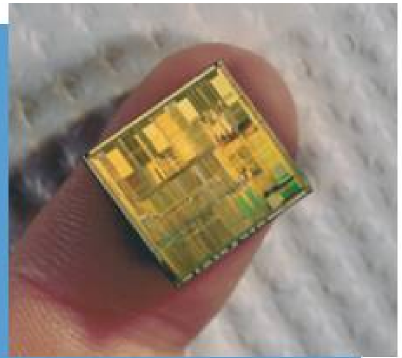
The Pentium® Pro processor delivered more performance than previous generation processors through an innovation called Dynamic Execution. This made possible the advanced 3D visualization and interactive capabilities.



Intel® Pentium® II processor
Intel® Pentium II Xeon® processor
Introduced 1997
Initial clock speed
300 MHz
Number of transistors
7,500,000
Manufacturing technology
0.25µ

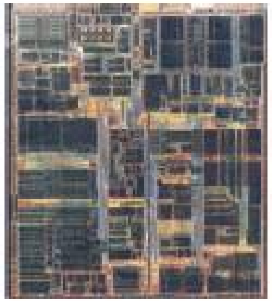
The Intel® Pentium® II processor's significant performance improvement over previous Intel-Architecture processors was based on the seamless combination of the P6 microarchitecture and Intel MMX media enhancement technology.

1994
Intel chips powered almost 75 percent of all desktop computers.



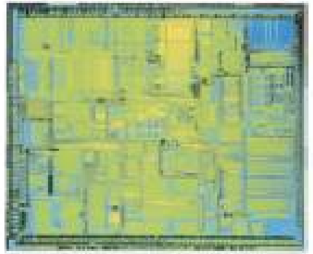
1995
Released in the fall of 1995, the Intel® Pentium® Pro processor was designed to fuel 32-bit server and workstation applications, enabling fast computer-aided design, mechanical engineering and scientific computation.

1998
The Intel® Pentium II Xeon processors feature technical innovations specifically designed for workstations and servers that utilize demanding business applications.



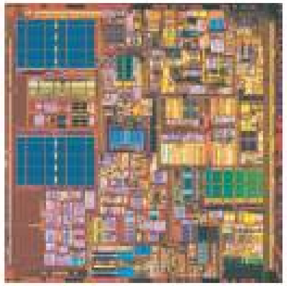
Intel® Pentium® II processor
Intel® Pentium II Xeon® processor
Introduced 1997
Initial clock speed
300 MHz
Number of transistors
7,500,000
Manufacturing technology
0.25µ

The Intel® Pentium® II processor's significant performance improvement over previous Intel-Architecture processors was based on the seamless combination of the P6 microarchitecture and Intel MMX media enhancement technology.



Intel® Pentium® III processor
Intel® Pentium® III Xeon® processor
Introduced 1999
Initial clock speed
500 MHz
Number of transistors
9,500,000
Manufacturing technology
0.18µ

The Intel® Pentium® III processor executed Internet Streaming SIMD Extensions, extended the concept of processor identification and utilized multiple low-power states to conserve power during idle times.

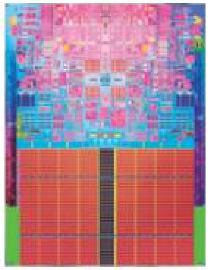


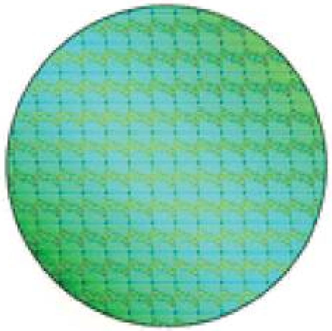
Intel® Pentium® 4 processor
Introduced 2000
Intel® Xeon® processor
Introduced 2001
Initial clock speed
1.5 GHz
Number of transistors
42,000,000
Manufacturing technology
0.18µ

The Intel® Pentium® 4 processor ushers in the advent of the nanotechnology age.

2001
The Itanium® processor is the first in a family of 64-bit products from Intel and is designed for high-end, enterprise-class servers and workstations.

2003
Intel® Centrino® mobile technology brought high performance, enhanced battery life, and integrated WLAN capability to thinner, lighter PCs.





Intel® Core™ 2 Duo processor
Intel® Core™2 Extreme processor
Dual-Core Intel® Xeon® processor
Introduced 2006
Initial clock speed
2.93 GHz
Number of transistors
291,000,000
Manufacturing technology
65nm

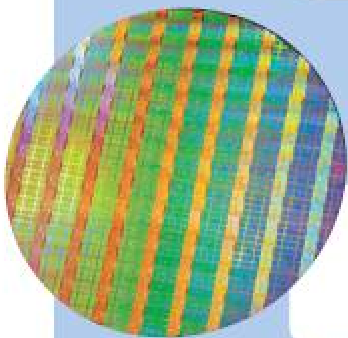
Intel® Core™2 Duo processor optimizes mobile microarchitecture of the Intel® Pentium® M processor and enhanced it with many microarchitecture innovations. Intel® Centrino® Pro and Intel® vPro™ processor technology provide excellent performance from the Dual-Core Intel® Core™2 Duo processor.

Dual-Core Intel® Itanium® 2 processor 9000 series
Introduced 2006
Initial clock speed
1.66 GHz
Number of transistors
1,720,000,000
Manufacturing technology
90nm

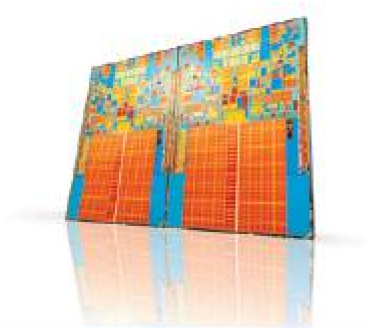
Dual-Core Intel® Itanium® 2 processor 9000 series outperforms the earlier, single-core version of the Itanium 2 processors. With more than 1.7 billion transistors and with two execution cores, these processors double the performance of previous Itanium processors while reducing average power consumption.

2005
Dual-core technology was introduced.

2006
Intel launched four processors for servers under the Xeon 5300 brand, and another processor under the Core 2 Extreme series for high performance computing. These "quad-core" processors show improved performance over others with just one or two processing cores.



2007
In the second half of 2007, Intel began production of the next generation Intel® Core™2 and Xeon processor families based on 45-nanometer (nm) Hi-k metal gate silicon technology.



Quad-Core Intel® Xeon® processor
Quad-Core Intel® Core™2 Extreme processor
Introduced 2006
Intel® Core™2 Quad processors
Introduced 2007
Initial clock speed
2.66 GHz
Number of transistors
582,000,000
Manufacturing technology
65nm

The unprecedented performance of the Intel® Core™2 Quad processor is made possible by each of the four complete execution cores delivering the full power of Intel Core microarchitecture. The Quad-Core Intel® Xeon® processor provides 50 percent greater performance than industry-leading Dual-Core Intel® Xeon® processor in the same power envelope. The quad-core-based servers enable more applications to run with a smaller footprint.

Quad-Core Intel® Xeon® processor (Penryn)
Dual-Core Intel® Xeon® processor (Penryn)
Quad-Core Intel® Core™2 Extreme processor (Penryn)
Introduced 2007
Initial clock speed
> 3 GHz
Number of transistors
820,000,000
Manufacturing technology
45nm

Intel's next generation Intel® Core™2 processor family, codenamed "Penryn", contains industry-leading microarchitecture enhancements. Further, new SSE4 instructions for improved video, imaging, and 3D content performance and new power management features will extend "Penryn" processor family leadership in performance and energy efficiency.

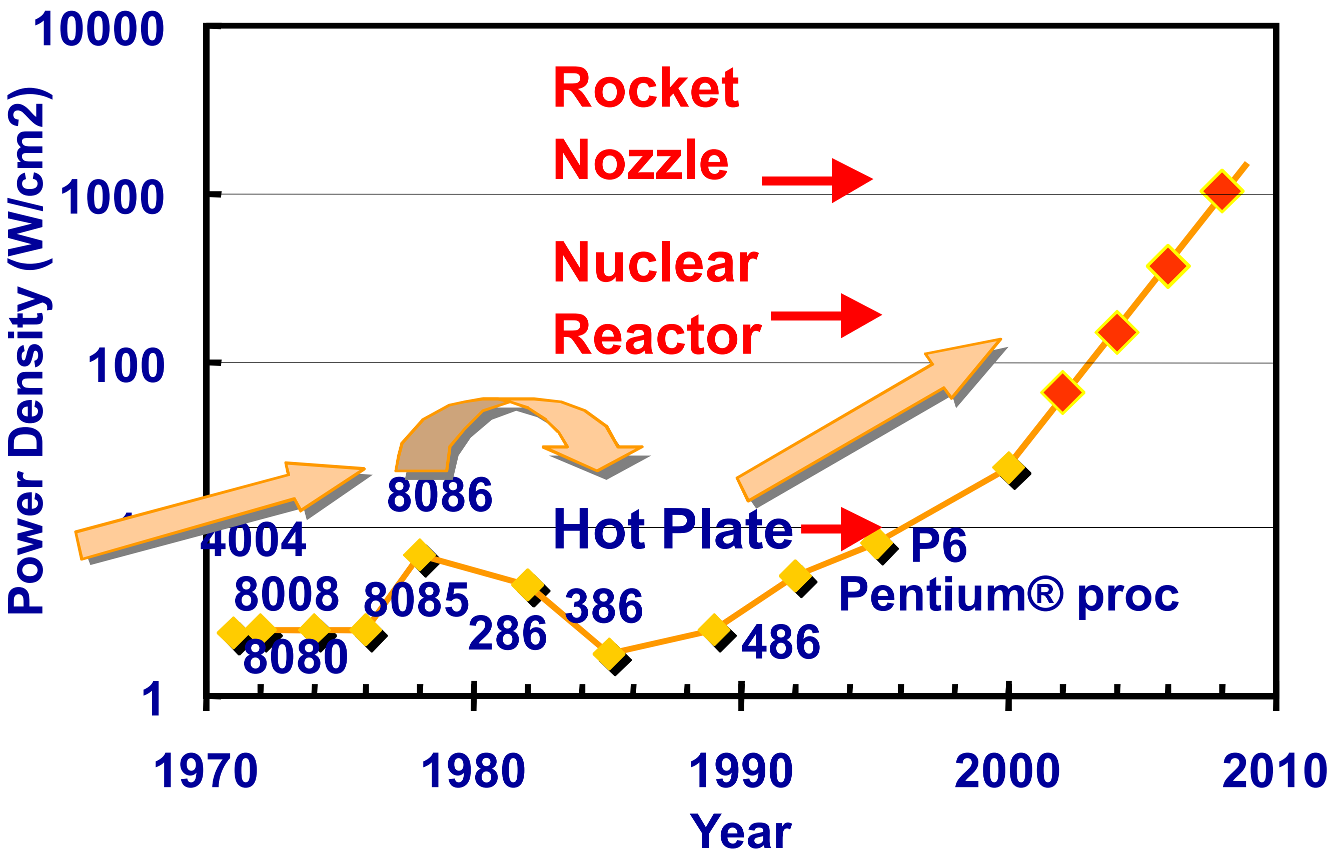
Evolution of Intel Microprocessors

Processor	Codename	Year Introduced	Transistors	Minimum Feature Size (microns)	Package	Socket or Slot	Core/Bus Frequency (Max)[1]	External Data Bus Width	Internal Register Widths	Address Bus Width	NDP[2]	L1 Cache	L2 Cac
4004		1971	2,250	10.0	16 pin DIP		.108 MHz	4	8	12	none	none	no
8008		1972	3,500	10.0	18 pin DIP		.200 MHz	8	8	14	none	none	no
8080		1974	6,000	6.0	40 pin DIP		3 MHz	8	8	16	none	none	no
8085[3]		1976	6,000	6.0	40 pin DIP		6 MHz	8	8	16	none	none	no
8086		1978	29,000	3.0	40 pin DIP		10 MHz	16	16	20	external	none	no
8088		1979	29,000	3.0	40 pin DIP		10 MHz	8	16	20	external	none	no
80286		1982	134,000	1.5	68 pin PLCC or PGA[4]		12.5 MHz	16	16	24	external	none	no
80386DX		1985	275,000	1.0	132 pin PGA or QFP[5]		33 MHz	32	32	32	external	none	exter
80386SX		1988	275,000	1.0	100 pin PQFP[7]		33 MHz	16	32	24	external	none	exte
80486DX		1989	1.2 million	0.8	168 pin PGA	Socket 3	50 MHz	32	32	32	on-chip	8 KB	exte
80486SX		1991	1.185 million	1.0	196 lead PQFP or 168 pin PGA	Socket 3	33 MHz	32	32	32	none	8 KB	exte
80486DX2		1992	1.2 million	0.6	168 pin PGA	Socket 3	66/33 MHz	32	32	32	on-chip	8 KB	exte
80486DX4		1994	1.2 million	0.6	168 pin PGA	Socket 3	100/ 33 MHz	32	32	32	on-chip	8 KB	exte
Pentium Classic	P5	1993	3.1 million	0.8	273 pin PGA	Socket 4, 5	66 MHz	64	32	32	on-chip	8/8 KB C/D[8]	exte
Pentium Classic	P54	1994	3.3 million	0.35, 0.5	296 pin PGA	Socket 7	200/66 MHz	64	32	32	on-chip	8/8 KB C/D	exte
Pentium MMX	P55	1997	4.5 million	0.25, 0.28	296 pin PGA	Socket 7	300/66 MHz	64	32	32	on-chip	16/16 KB C/D	exte
Pentium Pro	P6	1995	5.5 million[9]	0.35, 0.5	387 pin dual cavity PGA or PPGA[10]	Socket 8	200/66 MHz	64	32	36	on-chip	8/8 KB C/D	256/ 1M

Pentium II	(Klamath) Deschutes[12]	(1997) 1998	7.5 million	(0.28), (0.25)	242 contact SEC cartridge	Slot 1	(233/66 MHz) 450/100 MHz	64	32	36	on-chip	16/16 KB C/D	512 KB[13]
Celeron	(Covington) Mendocino[14]	1998	(7.5 million) 19 million[15]	0.25	(242 contact SEP cartridge)	Slot 1	(300/66 MHz)						
					370 pin PPGA	Socket 370	466/66 MHz	64	32	36	on-chip	16/16 KB C/D	(external) 128 KB[16]
Pentium III	Katmai	1999	9.5 million	0.25	242 contact SEC cartridge	Slot 1	550/100 MHz	64	32	36	on-chip	16/16 KB C/D	512 KB[17]
					330 contact SEC cartridge	Slot 2							
	Coppermine	1999		0.18	370 pin PGA	Socket 370	733/133 MHz						256 KB[18]
Itanium[19]	Merced	2000		0.18			6XX/133 MHz	128	64	64	on-chip		256 KB[20]

[1]It is likely that higher frequency versions of the newer processors will be offered in the future.
[2]Numeric data processor (also called coprocessor or floating point unit).
[3]Improved 8080 with three new instructions to enable/disable three added interrupt pins. Simplified hardware with single +5 V power supply and on-board clock generator.
[4]Plastic leaded chip carrier or pin grid array.
[5]Quad flat package (QFP).
[6]Some 386 computers (and nearly all later processors) incorporated external L2 caches.
[7]Plastic quad flat package.
[8]Separate code and data caches are supplied
[9]On-board 256 KB L2 cache (separate silicon die) has 15.5 million transistors (31 million for 512 KB cache). 1 MB cache has two separate 512 KB die.
[10]Plastic pin grid array
[11]Separate die in package. Cache operates at core speed.
[12]Specifications for Klamath processor are shown in parentheses.
[13]Separate die in SEC package. Cache operates at one-half core speed.
[14]Specifications for the Covington processor are shown in parentheses. The Mendocino processor is also called Celeron A.
[15]Includes integrated 128 KB L2 cache.
[16]128 KB cache is on the same die with the processor and operates at the core frequency of the processor.
[17]Separate die operating at 0.5 times core speed (slot 1) or integrated with the processor operating at core speed (slot 2).
[18]Integrated with the processor and operating at core speed. Includes 256-bit (vs. 64 bit on previous chips) processor–cache data bus.
[19]Specifications for this processor have not yet been finalized by Intel.
[20]Integrated with the processor die and operating at full core speed.

Power Density



Courtesy, Intel

Power Density increase

Power Density

Evolution in terms of Technology

Year	1947	1950	1961	1966	1971	1980	1990	2000
Technology	*Invention of the transistor*	*Discrete components*	*SSI*	*MSI*	*LSI*	*VLSI*	*ULSI**	*GSI†*
Approximate numbers of transistors per chip in commercial products	1	1	10	100–1000	1000–20,000	20,000–1,000,000	1,000,000–10,000,000	>10,000,000
Typical products	—	Junction Transistor and diode	Planar devices Logic gates Flip-flops	Counters Multiplexers Adders	8 bit micro-processors ROM RAM	16 and 32 bit micro-processors Sophisticated peripherals GHM Dram	Special processors, Virtual reality machines, smart sensors	

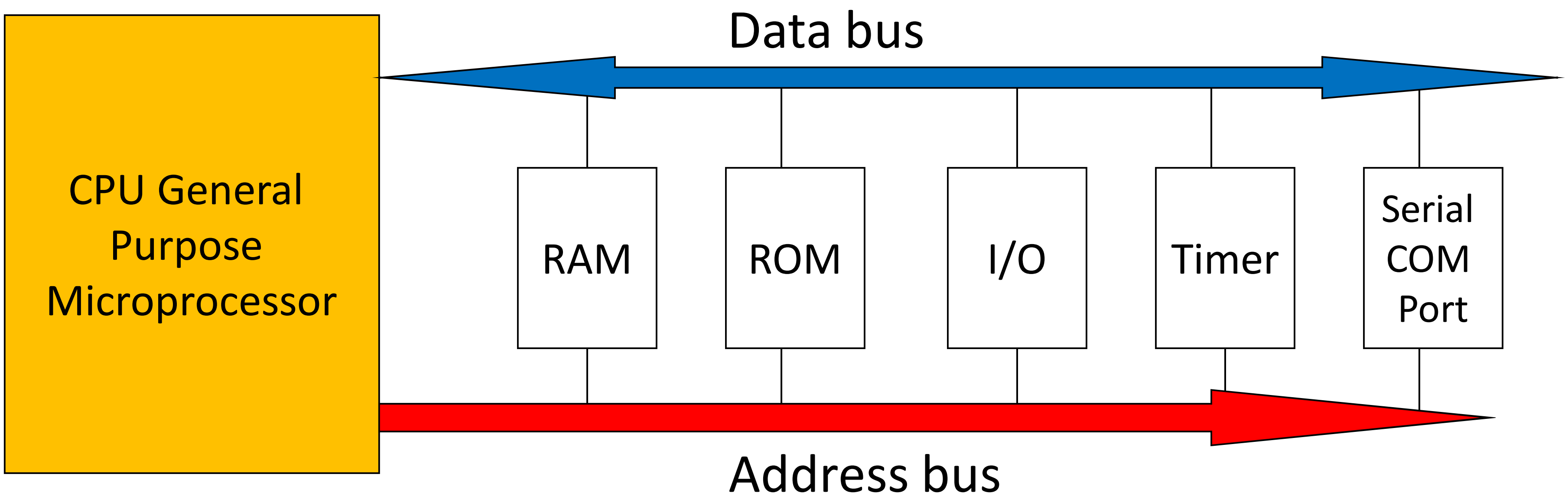
* Ultra large-scale integration
† Giant-scale integration

Types of Microcomputers

- *Microprocessor:* Processor on a chip
- In 1982, IBM began selling the idea of a *personal computer*. It featured a system board designed around the Intel 8088 8-bit microprocessor, 16 K memory and 5 expansion slots.
  - This last feature was the most significant one as it opened the door for 3rd party vendors to supply video, printer, modem, disk drive, and RS 232 serial adapter cards.
  - Generic PC: A computer with interchangable components manufactured by a variety of companies
- *Microcontroller* is an entire computer on a chip, a microprocessor with on-chip memory and I/O.
  - These parts are designed into (embedded within) a product and run a program which never changes
  - Home appliances, modern automobiles, heat, air-conditioning control, navigation systems
  - Intel's MCS-51 family, for example, is based on an 8-bit microprocessor, but features up to 32K bytes of on-board ROM, 32 individually programmable digital input/output lines, a serial communications channel.

General Purpose Microprocessors

Microprocessors lead to versatile products



These general microprocessors contain no RAM, ROM, or I/O ports on the chip itself

Ex. Intel's x86 family (8088, 8086, 80386, 80386, 80486, Pentium)

Motorola's 680x0 family (68000, 68010, 68020, etc)

Microcontrollers

Microcontroller

CPU	RAM	ROM
I/O	TIMER	Serial Com Port

A microcontroller has a CPU in addition to a fixed amount of RAM, ROM, I/O ports on one single chip; this makes them ideal for applications in which cost and space are critical

Example: a TV remote control does not do computing power of a 486

Embedded Systems

- An embedded system uses a microcontroller or a microprocessor to do one task and one task only
  - Example: toys, garage door openers, answering machines, ABS, keyless entry, etc.
  - Inside every mouse, there is a microcontroller that performs the task of finding the mouse position and sends it to the PC
- Although microcontrollers are the preferred choice for embedded systems, there are times that the microcontroller is inadequate for the task
- Intel, Motorola, AMD, Cyrix have also targeted the embedded market with their general purpose microprocessors
- For example, Power PC microprocessors (IBM Motorola joint venture) are used in PCs and routers/switches today
- Microcontrollers differ in terms of their RAM,ROM, I/O sizes and type.
  - ROM: One time-programmable, UV-ROM, flash memory

Instruction Set

- The list of all recognizable instructions by the instruction decoder is called the instruction set
  - CISC (Complex Instruction Set Computers), e.g., 80x86 family has more than 3000 instructions
  - RISC (Reduced Instruction Set Computers) - A small number of very fast executing instructions
- Most microprocessor chips today are allowed to fetch and execute cycles to overlap
  - This is done by dividing the CPU into
    - EU (Execution Unit)
    - BIU (Bus Interface Unit)
  - BIU fetches instructions from the memory as quickly as possible and stores them in a queue, EU then fetches the instructions from the queue not from the memory
    - The total processing time is reduced
  - Modern microprocessors also use a *pipelined* execution unit which allows the decoding and execution of instructions to be overlapped.

RISC versus CISC

- **Advantages of complex instruction set machines (CISC)**
  - Less expensive due to the use of microcode; no need to hardwire a control unit
  - Upwardly compatible because a new computer would contain a superset of the instructions of the earlier computers
  - Fewer instructions could be used to implement a given task, allowing for more efficient use of memory
  - Simplified compiler, because the microprogram instruction sets could be written to match the constructs of high-level languages
  - More instructions can fit into the cache, since the instructions are not a fixed size
- **Disadvantages of CISC**

Although the CISC philosophy did much to improve computer performance, it still had its drawbacks:

  - Instruction sets and chip hardware became more complex with each generation of computers, since earlier generations of a processor family were contained as a subset in every new version
  - Different instructions take different amount of time to execute due to their variable-length
  - Many instructions are not used frequently; Approximately 20% of the available instructions are used in a typical program

RISC versus CISC

Advantages of RISC

Advantages of a reduced instruction set machine:

- Faster
- Simple hardware
- Shorter design cycle due to simpler hardware

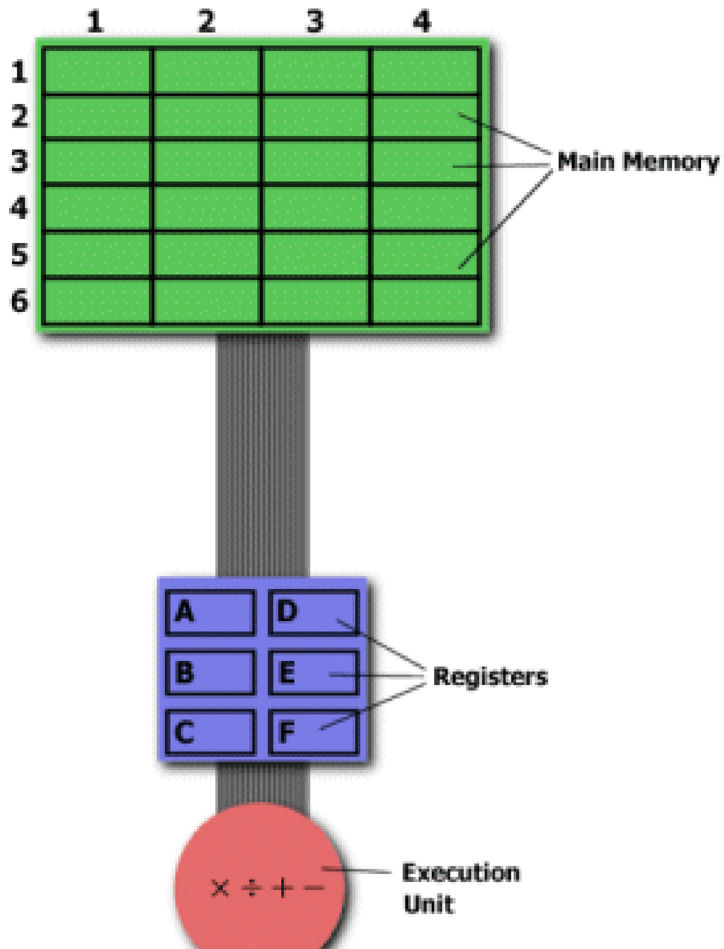
Disadvantages of RISC

Drawbacks of a reduced instruction set computer include

- Programmer must pay close attention to instruction scheduling so that the processor does not spend a large amount of time waiting for an instruction to execute
- Debugging can be difficult due to the instruction scheduling Require very fast memory systems to feed them instructions

•Nearly all modern microprocessors, including the Pentium (hybrid RISC/CISC) Power PC, Alpha and SPARC microprocessors are superscalar

More on RISC and CISC



SOURCE: ARSTECHNICA

LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A

MULT 2:3, 5:2

What happens when you turn on a PC in general

- The process of bringing up the operating system is called *booting*
- Your computer knows how to boot because instructions for booting are built into one of its chips, the BIOS (or Basic Input/Output System) chip.
- The BIOS chip tells it to look in a fixed place, usually on the lowest-numbered hard disk (the *boot disk*) for a special program called a *boot loader* (under Linux this is LILO).
- The boot loader is pulled into memory and started. The boot loader's job is to start the real operating system.
- The loader does this by looking for a *kernel*, loading it into memory, and starting it.
- Once the kernel starts, it has to look around, find the rest of the hardware, and get ready to run programs.
- The kernel’s first job is usually to check to make sure your disks are OK.
- Then kernel starts several *daemons*. A daemon is a program like a print spooler, a mail listener or a WWW server that lurks in the background, waiting for things to do.
- Finally an interaction with the user is initiated.

Computer Operating Systems

- What happens when the computer is first turned on?
- MS-DOS
  - At interrupt location FFFF:0000 there is a link to a startup program in the BIOS.
  - This program in turn accesses the master boot record on the floppy or hard disk drive
  - A loader then transfers the system files IO.SYS and MSDOS.SYS from the disk drive to the main memory
  - Finally, the command interpreter COMMAND.COM is loaded into memory which puts the DOS prompt on the screen that gives the user access to DOS's built-in commands like DIR, COPY, VER.
- The 640 K Barrier
  - DOS was designed to run on the original IBM PC
  - 8088 microprocessor, 1Mbytes of main memory
  - IBM divided this 1Mb address space into specific blocks
    - 640 K of RAM (user RAM)
    - 384 K reserved for ROM functions (control programs for the video system, hard drive controller, and the basic input/output system)

MS-DOS Functions and BIOS Services

- Program Support
  - *BIOS:* usually stored in ROM these routines provide access to the hardware of the PC
  - Access to the BIOS is done through the software interrupt instruction Int *n*
  - For example, the BIOS keyboard services are accessed using the instruction INT 16h
  - In addition to BIOS services DOS also provides higher level functions
    - INT 21h
    - More details later

memory map of the IBM PC

- The 20-bit address of 8088/86 allows 1mb (1024K bytes) of memory space with the address range 00000–FFFFF.
  - During the design phase of the first IBM PC, engineers had to decide
    on the allocation of the 1-megabyte memory space to various sections
    of the PC.
    - This memory allocation is called a *memory map*.

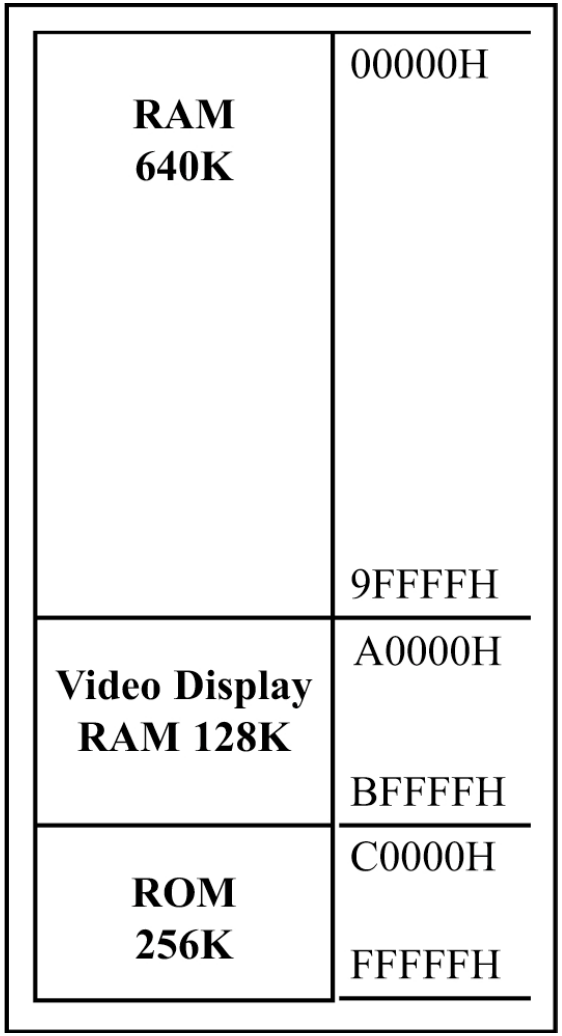


Figure 1-3 Memory Allocation in the PC

memory map of the IBM PC

- Of this 1 megabyte, 640K bytes from addresses **00000–9FFFFH** were set aside for RAM
- 128K bytes **A0000H– BFFFFH** were allocated for video memory
- The remaining 256K bytes from **C0000H–FFFFFH** were set aside for ROM

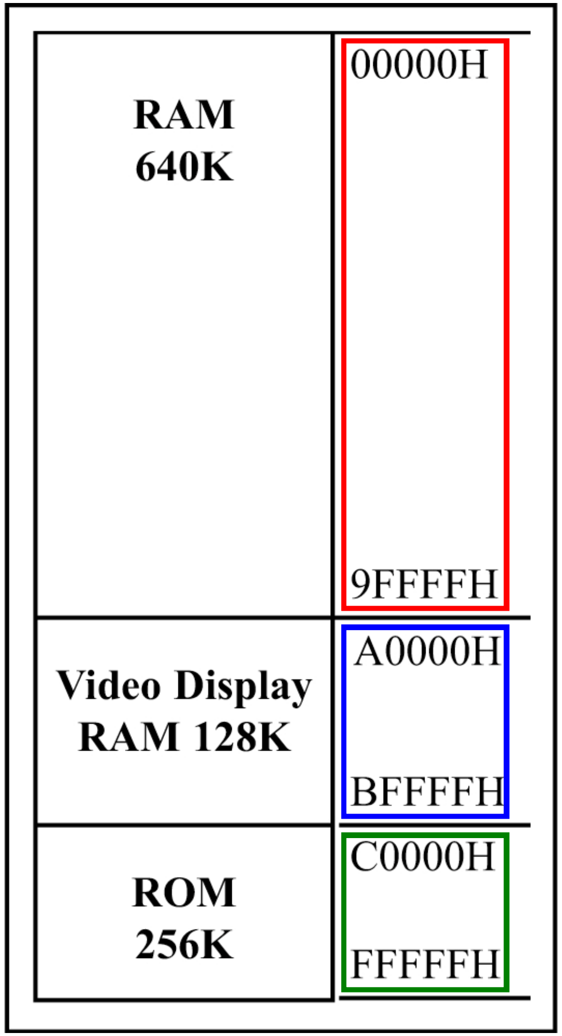
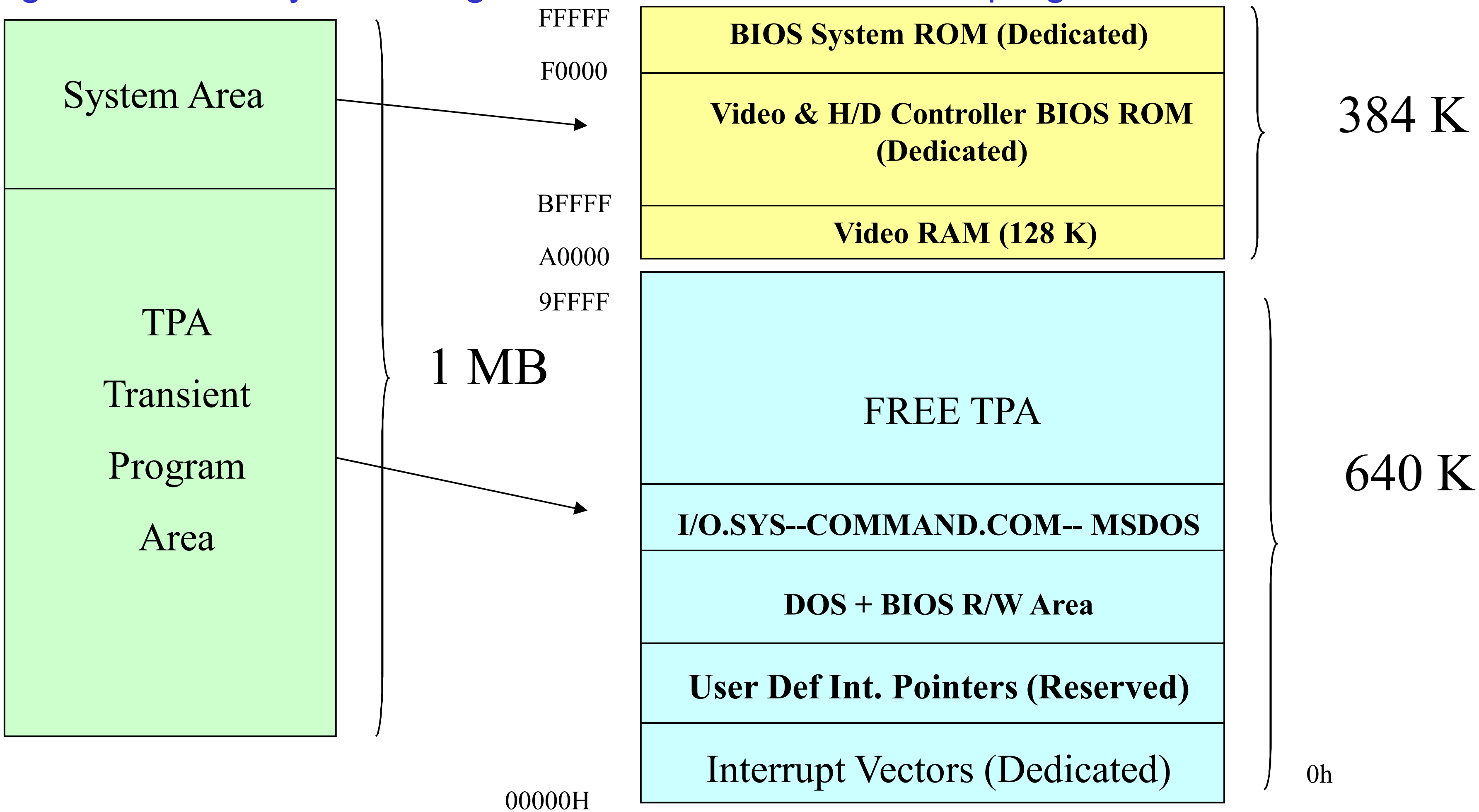


Figure 1-3 Memory Allocation in the PC

Dedicated, Reserved and General Purpose Memory

- Some address locations have dedicated functions and should not be used as general memory for storage of data or instructions of a program



Brey 19
Mazidi 32

more about RAM

- In the early 80s, most PCs came with 64K to 256K bytes of RAM, more than adequate at the time
  - Users had to buy memory to expand up to 640K.
- Managing RAM is left to Windows because...
  - The amount of memory used by Windows varies.
  - Different computers have different amounts of RAM.
  - Memory needs of application packages vary.
- For this reason, we do not assign any values for the CS, DS, and SS registers.
  - Such an assignment means specifying an exact physical address in the range 00000–9FFFFH, and this is beyond the knowledge of the user.

video RAM

- From A0000H to BFFFFH is set aside for video
  - The amount used and the location vary depending on the video board installed on the PC

more about ROM

- C0000H to FFFFFH is set aside for ROM.
  - Not all the memory in this range is used by the PC's ROM.
- 64K bytes from location F0000H–FFFFFH are used by BIOS (basic input/output system) ROM.
  - Some of the remaining space is used by various adapter cards (such as the network card), and the rest is free.
- The 640K bytes from 00000 to 9FFFFH is referred to as *conventional memory*.
  - The 384K bytes from A0000H to FFFFFH are called the UMB (*upper memory block*).

function of BIOS ROM

- There must be some permanent (nonvolatile) memory to hold the programs telling the CPU
  what to do when the power is turned on
  - This collection of programs is referred to as BIOS.
- BIOS stands for *basic input-output system*.
  - It contains programs to test RAM and other components connected to the CPU.
  - It also contains programs that allow Windows to communicate with peripheral devices.
  - The BIOS tests devices connected to the PC when the computer is turned on and to report any errors.

Some Important Terminology

- Bit is a binary digit that can have the value 0 or 1
- A byte is defines as 8 bits
- A nibble is half a byte
- A word is two bytes in general or it is the number of bits it can handle at one time. For example: word size is 8! Or word size is 16.
- A double word is four bytes
- A kilobyte is 2^10 bytes (1024 bytes), The abbreviation K is most often used
  - Example: A floppy disk holding 356Kbytes of data
- A megabyte or meg is 2^20 bytes, it is exactly 1,048,576 bytes
- A gigabyte is 2^30 bytes
- 1 Terabyte is 2^40 bytes etc

Inside the Computer - Internal organization of computers

- Internal workings of every computer can be broken down into three parts:

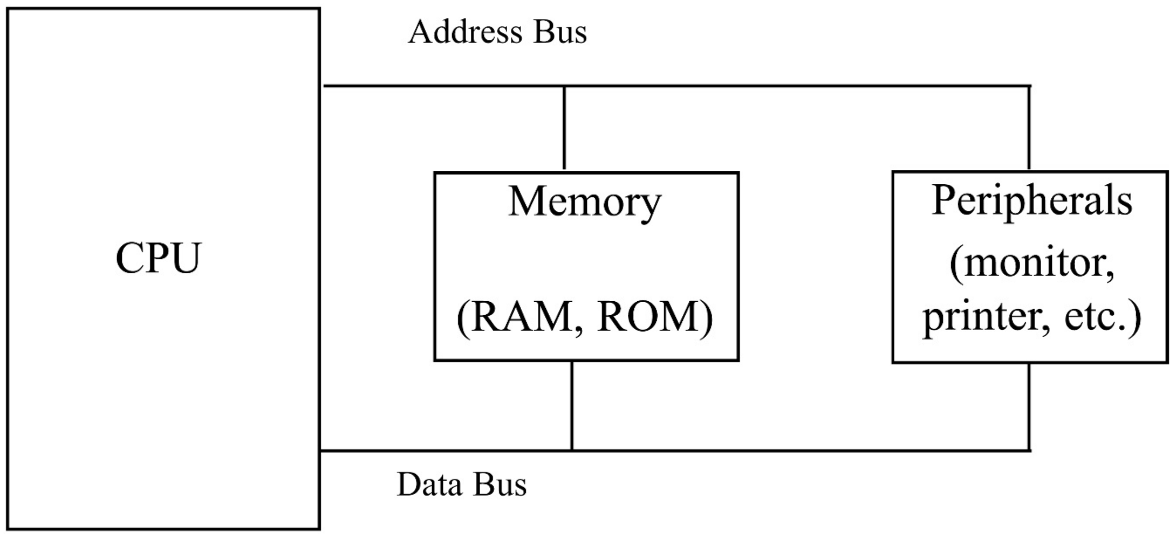


Figure 0-9
Inside the
Computer

Inside the Computer - Internal organization of computers

- Internal workings of every computer can be broken down into three parts:
  - **CPU** (central processing unit).

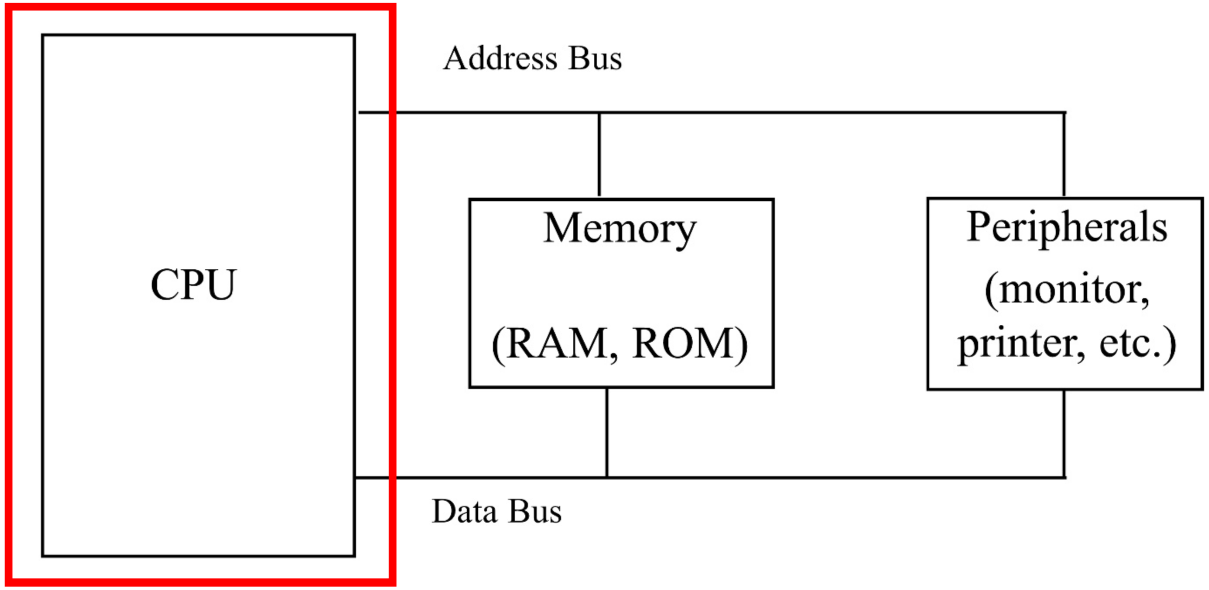
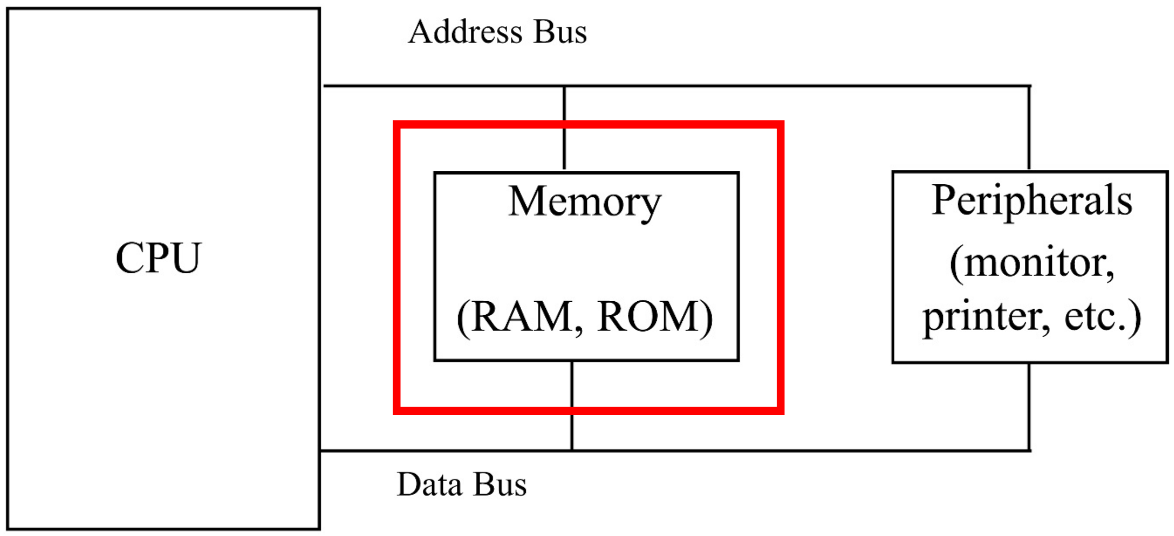


Figure 0-9
Inside the Computer

Inside the Computer - Internal organization of computers

- Internal workings of every computer can be broken down into three parts:
  - **CPU** (central processing unit).
  - **Memory**.

Figure 0-9 Inside the Computer



Inside the Computer - Internal organization of computers

- Internal workings of every computer can be broken down into three parts:
  - **CPU** (central processing unit).
  - **Memory**.
  - **I/O** (input/output) devices.

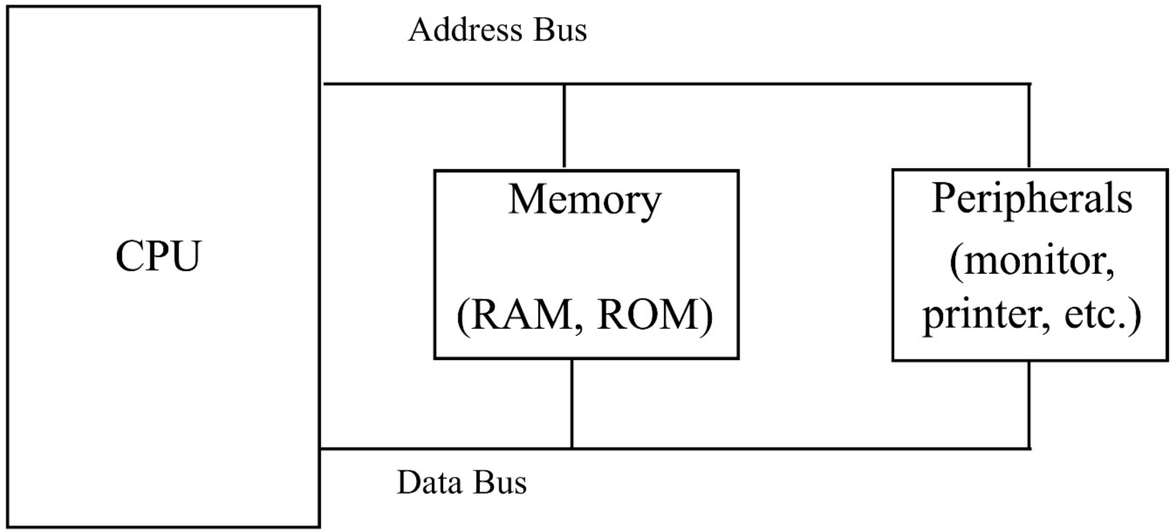


Figure 0-9
Inside the Computer

Inside the Computer - Internal organization of computers

- CPU function is to execute (process) information stored in memory.
- I/O devices, such as keyboard & monitor provide a means of communicating with the CPU.
- The CPU is connected to memory and I/O through a group of wires called a *bus*.
  - Allows signals to carry information from place to place.
- In every computer there are three types of buses:
  - Address bus; Data bus; Control bus.

Inside the Computer - Internal organization of computers

- For a device (memory or I/O) to be recognized by the CPU, it must be assigned an address.
  - No two devices can have the same address.
    - The address assigned to a given device must be unique.
- The CPU puts the address (in binary) on the address bus & decoding circuitry finds the device.
- The CPU then uses the data bus either to get data from that device or to send data to it.
- Control buses provide device read/write signals to indicate if the CPU is asking for, or sending information.

Three Bus System Architecture

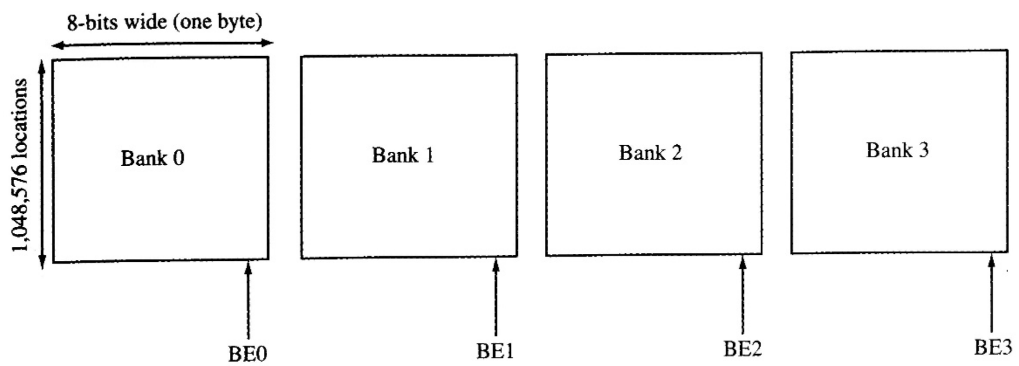
- A collection of electronic signals all dedicated to particular task is called a *bus*
  - *data bus*
  - *address bus*
  - *control bus*
- **Data Bus**
  - The width of the data bus determines how much data the processor can read or write in one memory or I/O cycle (**Machine Cycle)**
  - 8-bit microprocessor has an 8-bit data bus
  - 80386SX 32-bit internal data bus, 16-bit external data bus
  - 80386 32-bit internal and external data busses
  - **8086 has 16 bits for data where 8088 has 8 bits for data externally however they both work 16 bits internally**
  - Data Buses are bidirectional.
  - More data means more expensive computer however faster processing speed.

Inside the Computer – Data Bus

- As data buses carry information in/out of a CPU, the more data buses available, the better the CPU.
  - More buses mean a more expensive CPU & computer.
- Data buses are bidirectional, because the CPU must use them either to receive or to send data.
  - Average bus size is between 8 and 64.

- Computer processing power is related to bus size.
  - An 8-bit bus can send out 1 byte a time.
  - A 16-bit bus can send out 2 bytes at a time.
    - Twice as fast!

Address Bus



Here the Total amount of memory is 4Mbytes

- **Address Bus** - Unidirectional
  - The address bus is used to identify the memory location or I/O device (also called port) the processor intends to communicate with
  - 20 bits for the 8086 and 8088
  - 32 bits for the 80386/80486 and the Pentium
  - 36 bits for the Pentium Pro
- 8086 has a 20-bit address bus and therefore addresses all combinations of addresses from all 0s to all 1s. This corresponds to 2^{20} addresses or 1M (1 Meg) addresses or memory locations.
- Pentium: 4Gbyte main memory

Inside the Computer – Address Bus

- The address bus is used to identify devices and memory connected to the CPU.
  - The more address bits available, the larger the number of devices that can be addressed.
- The number of CPU address bits determines the number of locations with which it can communicate.
  - Always equal to 2^x, where x is the number of address lines, regardless of the size of the data bus.
- The address bus is *unidirectional.*
  - The CPU uses the bus only to send addresses *out.*

Control Bus

- Control bus is Uni-directional
- How can we tell the address is a memory address or an I/O port address
  - Memory Read
  - Memory Write
  - I/O Read
  - I/O Write
- When Memory Read or I/O Read are active, data is *input* to the processor.
- When Memory Write or I/O Write are active, data is *output* from the processor.
- The control bus signals are defined from the processor's point of view.

Inside the Computer - Internal organization of computers

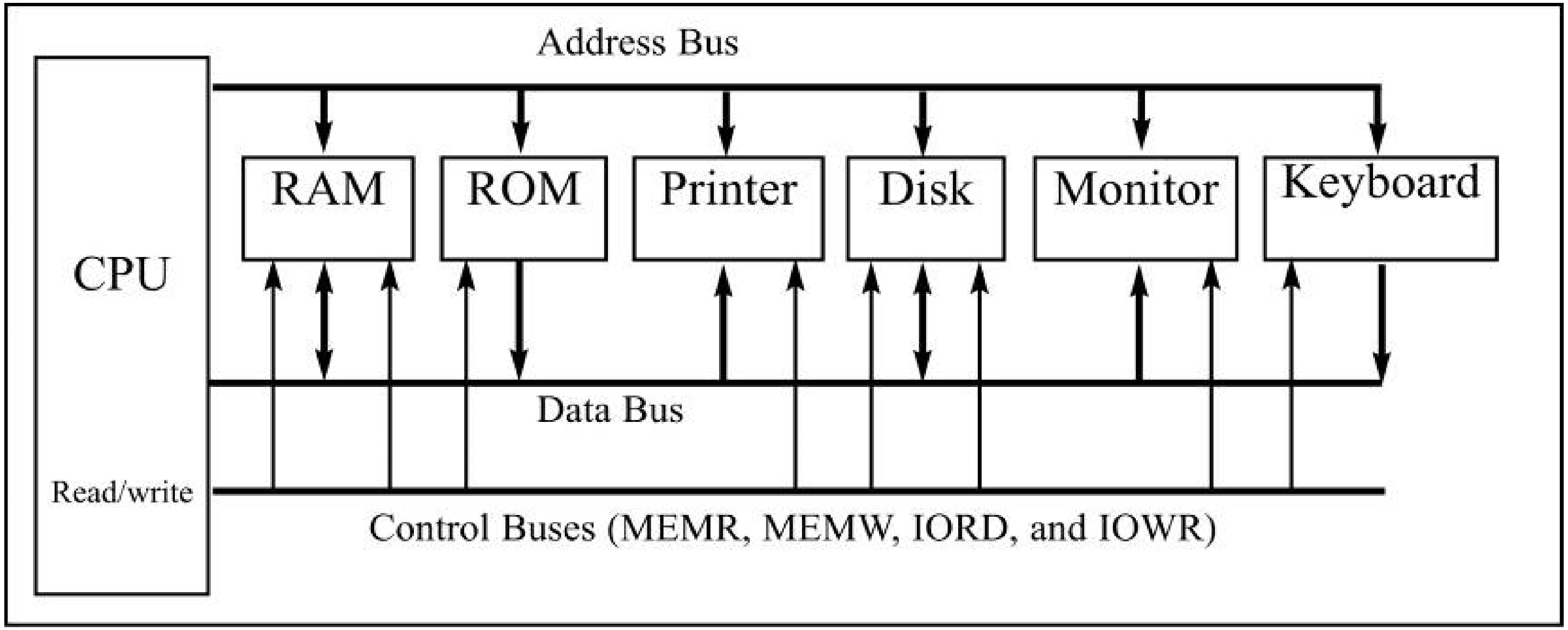


Figure 0-10 Internal Organization Of Computers

Inside the Computer - Internal organization of computers

- For the CPU to process information, the data must be stored in RAM or ROM.
  - The CPU cannot get the information from the disk directly because the disk is too slow.
  - RAM & ROM are often referred to as *primary memory*.
    - Disks are called *secondary memory*.
- ***RAM*** **-** which stands for "random access memory" (sometimes called *read/write memory*).
  - Used for temporary storage of programs while running.
    - Data is lost when the computer is turned off.
    - RAM is sometimes called *volatile memory.*
- ***ROM*** **-** stands for "read-only memory".
  - Contains programs and information essential to the operation of the computer.
    - Information in ROM is permanent, cannot be changed by the user, and is not lost when the power is turned off.
    - ROM is called *nonvolatile memory.*

Inside the Computer - Internal organization of computers

- A program stored in memory provides instructions to the CPU to perform an action.
  - Adding payroll numbers or controlling a robot.

Function of the CPU is to fetch these instructions from memory and then execute them.

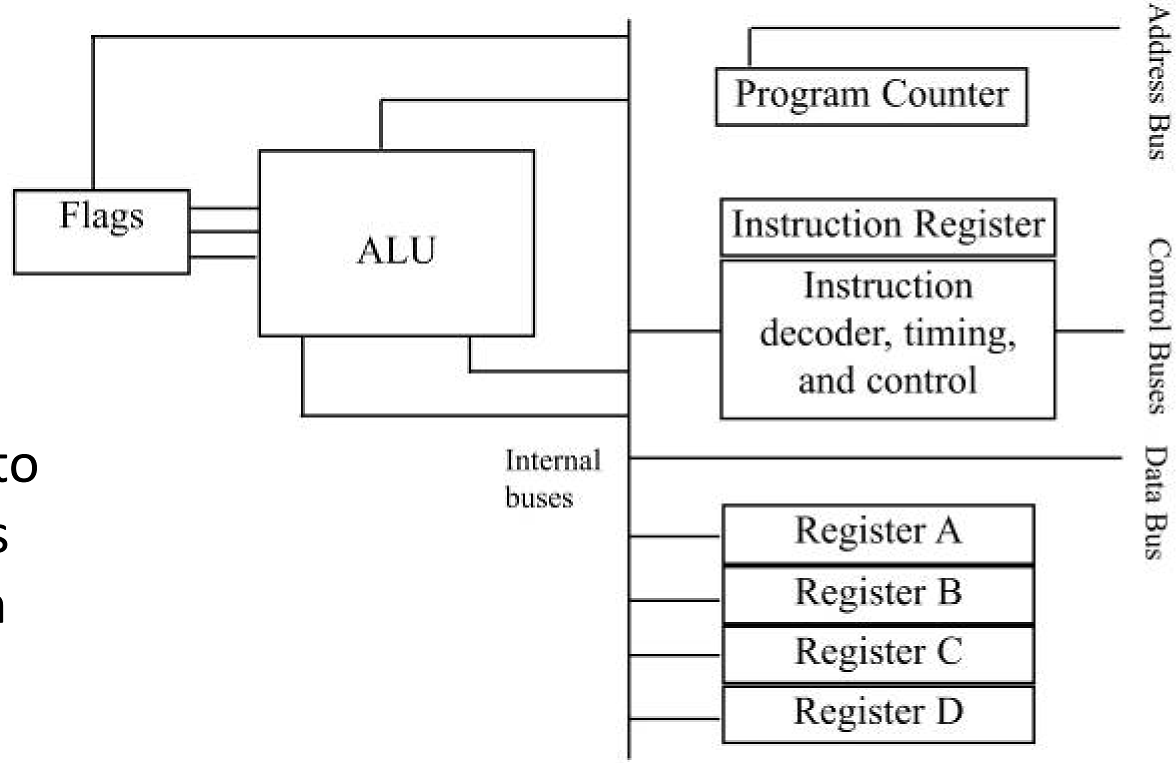


Figure 0-19 Internal Block Diagram of a CPU

Inside the Computer - Internal organization of computers

- To perform the actions of fetch and execute, all CPUs are equipped with resources such as…
  - ***Registers*** - to store information temporarily.
    - 8, 16, 32, 64 bit, depending on CPU.
  - ***ALU*** (arithmetic/logic unit) - for arithmetic functions such as add, subtract, multiply, and divide.
    - Also logic functions such as AND, OR, and NOT.
  - ***Program counter*** - to point to the address of the next instruction to be executed.
    - In the IBM PC, a register called IP or *instruction pointer.*
  - ***Instruction decoder*** - to interpret the instruction fetched into the CPU.

Inside the Computer - Internal organization of computers

- A step-by-step analysis of CPU processes to add three numbers, with steps & code shown.
  - Assume a CPU has registers A, B, C, and D.
    - An 8-bit data bus and a 16-bit address bus.
  - The CPU can access memory addresses 0000 to FFFFH.
    - A total of 10000H locations.

Action	Code	Data
Move value 21H into register A	B0H	21H
Add value 42H to register A	04H	42H
Add value 12H to register A	04H	12H

Inside the Computer - Internal organization of computers

- If the program to perform the actions listed above is stored in memory locations starting at 1400H, the following would represent the contents for each memory address location…

Memory address	***Contents of memory address***
1400	(B0)code for moving a value to register A
1401	(21)value to be moved
1402	(04)code for adding a value to register A
1403	(42)value to be added
1404	(04)code for adding a value to register A
1405	(12)value to be added
1406	(F4)code for halt

Inside the Computer - Internal organization of computers

- The CPU's program counter can have a value between 0000 and FFFFH.
  - The program counter must be set to the address of the first instruction code to be executed - 1400H.

Memory address	*Contents of memory address*
1400	(B0)code for moving a value to register A
1401	(21)value to be moved
1402	(04)code for adding a value to register A
1403	(42)value to be added
1404	(04)code for adding a value to register A
1405	(12)value to be added
1406	(F4)code for halt

Inside the Computer - Internal organization of computers

- The CPU puts the address 1400H on the address bus and sends it out.
  - Memory finds the location while the CPU activates the READ signal, indicating it wants the byte at 1400H.
    - The content (B0) is put on the data bus & brought to the CPU.

Memory address	***Contents of memory address***
1400	(B0)code for moving a value to register A
1401	(21)value to be moved
1402	(04)code for adding a value to register A
1403	(42)value to be added
1404	(04)code for adding a value to register A
1405	(12)value to be added
1406	(F4)code for halt

Inside the Computer - Internal organization of computers

- The CPU decodes the instruction B0 with the help of its instruction decoder dictionary.
  - Bring the byte of the next memory location into CPU Register A.

Memory address	*Contents of memory address*
1400	(B0)code for moving a value to register A
1401	(21)value to be moved
1402	(04)code for adding a value to register A
1403	(42)value to be added
1404	(04)code for adding a value to register A
1405	(12)value to be added
1406	(F4)code for halt

Inside the Computer - Internal organization of computers

- From memory location 1401H, the CPU fetches code 21H directly to Register A.
  - After completing the instruction, the program counter points to the address of the next instruction - 1402H.
    - Address 1402H is sent out on the address bus, to fetch the next instruction.

Memory address	***Contents of memory address***
1400	(B0)code for moving a value to register A
1401	(21)value to be moved
1402	(04)code for adding a value to register A
1403	(42)value to be added
1404	(04)code for adding a value to register A
1405	(12)value to be added
1406	(F4)code for halt

Inside the Computer - Internal organization of computers

- From 1402H, the CPU fetches code 04H.
  - After decoding, the CPU knows it must add the byte at the next address (1403) to the contents of register A.
    - After it brings the value (42H) into the CPU, it provides the contents of Register A, along with this value to the ALU to perform the addition.
    - Program counter becomes 1404, the next instruction address.

Memory address	***Contents of memory address***
1400	(B0) code for moving a value to register A
1401	(21) value to be moved
1402	(04) code for adding a value to register A
1403	(42) value to be added
1404	(04) code for adding a value to register A
1405	(12) value to be added
1406	(F4) code for halt

Inside the Computer - Internal organization of computers

- Address 1404H is put on the address bus and the code is fetched, decoded, and executed.
  - Again adding a value to Register A.
    - The program counter is updated to 1406H

Memory address	***Contents of memory address***
1400	(B0) code for moving a value to register A
1401	(21) value to be moved
1402	(04) code for adding a value to register A
1403	(42) value to be added
1404	(04) code for adding a value to register A
1405	(12) value to be added
1406	(F4) code for halt

Inside the Computer - Internal organization of computers

- The contents of address 1406 (HALT code) are fetched in and executed.
  - The HALT instruction tells the CPU to stop incrementing the program counter and asking for the next instruction.
    - Without HALT, the CPU would continue updating the program counter and fetching instructions.

Memory address	***Contents of memory address***
1400	(B0)code for moving a value to register A
1401	(21)value to be moved
1402	(04)code for adding a value to register A
1403	(42)value to be added
1404	(04)code for adding a value to register A
1405	(12)value to be added
1406	(F4)code for halt

Computer Programming

- Machine Language vs Assembly Language
  - Machine language or object code is the only code a computer can execute but it is nearly impossible for a human to work with
  - E4 27 88 C3 E4 27 00 D8 E6 30 F4 the object code for adding two numbers input from the keyboard
- When programming a microprocessor, programmers often use assembly language
  - This involves 3-5 letter abbreviations for the instruction codes (mnemonics) rather than the binary or hex object codes

Address	Hex Object Code				Mnemonics		Comment
					Op-Code	Operand	
0100	E4	27			IN	AL,27H	Input first number from port 27H and store in AL
0102	88	C3			MOV	BL,AL	Save a copy of register AL in register BL
0104	E4	27			IN	AL,27H	Input second number to AL
0106	00	D8			ADD	AL,BL	Add contents of BL to AL and store the sum in AL
0107	E6	30			OUT	30H,AL	Output AL to port 30H
0109	F4				HLT		Halt the computer

Source code

Edit, Assemble, Test, and Debug Cycle

- Using an *editor*, the source code of the program is created. This means selecting the appropriate instruction mnemonics to accomplish the task
- A compiler program which examines the source code file generated by the editor and determines the object code for each instruction in the program, is then run. In assembly language programming, this is called an *assembler* (MASM (Chapter 2 of the textbook, DEBUG: Appendix A of the textbook, etc.,)
- The object code produced by the computer is loaded into the target computer's memory and is then *run.*
- *Debugging:* locating and fixing the source of error
- High-level programming Languages
  - Basic, Pascal, C, C++